Routing protocol evaluation for the IoT

Requirement analysis and experiment design for large-scale test beds.

Lotte Steenbrink
lotte.steenbrink@haw-hamburg.de

ABSTRACT

To gather realistic knowledge about routing in the IoT, routing protocol research and evaluation must complement the primary use of simulation with the use of large-scale testbeds. In this paper, a testbed-based evaluation approach for routing protocols is presented, with a strong focus on IoT applications. This approach was designed to be modular and extensible, so as to allow adaption to the high variation in network characteristics in different IoT use cases. Using this approach as a base, routing protocols can be evaluated concerning their suitability for different IoT scenarios, and possibilities for improvements can be uncovered.

Keywords

IoT, routing, MANET, LLN, RPL, AODV, RIOT, test beds

Categories and Subject Descriptors

C.2.2 [**Network Protocols**]: Routing protocols; B.6.3 [**Design Aids**]: Simulation

1. INTRODUCTION

The Internet of Things (IoT) envisions autonomous communication between small computers installed in everyday objects or distributed across industrial facilities in order to advance human interaction as well as productivity and security. In 2014, the IoT reached the peak of Gartner's hype cycle for emerging technologies[1]. It is both a growing market and a thriving research field. One central aspect of IoT communication is routing: finding the best paths between nodes and towards sink nodes and gateways is crucial to ensure energy-efficient and smooth network operations. However, practical experience with IoT routing is sparse, and scientific evaluation of such environments is rare. Most routing protocol evaluations are simulation-based, and many of these evaluations have not been designed with the IoT in mind. This paper presents a testbed-based evaluation approach tailored to the IoT. The goal is to enable the evaluation of routing protocols which have been created for Low Power and Lossy Networks (LLNs) or Mobile Ad-hoc Networks (MANETs) with regard to their suitability for the IoT. Ultimately, the goal is to evaluate and confirm assumptions about the suitability of specific protocols for specific environments, and to facilitate the extension and optimization of routing protocols based on their performance in certain types of networks.

The remainder of this paper is organized as follows: First, the need for experimental work is highlighted in section 2. Then, the different domains and use cases that form the IoT are assessed in section 3. Based on these findings, experiment goals, design, realization and evaluation details are assessed in sections 4, 5 and 6. Finally, a conclusion of all findings as well as an outlook into future steps is provided by section 7.

1.1 Related work

Research on the foundations of routing protocol evaluation has been done for about two decades, and is increasingly focused on test beds and the IoT. [1] provides a summary of issues which should be considered when evaluating a routing protocol. Routing requirements for IoT-like scenarios of home and building automation, as well as urban LLNs are defined in [2], [3] and [4]. With decreasing hardware costs and increasing demand for alternatives to evaluation through simulation, testbed sites and testbed-based research are increasing. [5] presents the features and failings of different Wireless Sensor Network Testbeds, along with a requirement analysis for IoT-ready testbeds. **??** discusses advantages and caveats of testbed-based research and proposes Virtual Testbeds (VTBs) which combine physical, simulated and emulated components. Furthermore, work discussing routing protocols in "real world" scenarios using test beds is on the rise. [6] presents a performance study of the Routing Protocol for Low-Power and Lossy Networks (RPL) using a testbed of over 250 nodes.[7] discusses influences on transmission range in food monitoring use cases, in particular monitoring bananas during transport. results were achieved both through mathematical analysis as well as a simple testbed consisting of four nodes.

2. EXPERIMENTATION VS. SIMULATION

To date, most IoT, LLN, and MANET routing research has been conducted with the help of simulations. This has many benefits: it is cost-effective and widely available, requires low maintenance, makes experiments easily reproducible, enables experimentation with hundreds of thousands of nodes, and provides an environment which can be controlled and monitored in detail: If packets are lost, for example, the cause on layer 0 can be examined in detail, and protocols can be optimized accordingly. However, simulations are always just a simplification of reality. Without "real life" data to check against, the accuracy of a simulation model cannot be determined. It has been shown that the assumptions made during network simulation often don't hold in the real world, which influences simulation results significantly [8]. Matters are complicated further by the fact

[1] `https://www.gartner.com/doc/2809728`

that that especially wireless networks are disturbed by outside influences such as moving objects, reflection or outside noise. This effect is magnified in LLNs, which can often be found in the IoT [9]. This can impact the performance of a network severely and is very hard to model in a simulation. The absence of these side effects can be of great benefit when studying specific traits of a protocol, but to determine a protocol's compatibility with the real world, its performance under such disturbances must be assessed, too. Consequently, it is necessary to obtain more realistic testing experience with the help of a growing fleet of test beds, made possible by technological progress and dedicated effort.

3. IOT DOMAINS AND USE CASES

By its very nature, the IoT encompasses a broad spectrum of environments and use cases. Surveys divide the IoT into domains such as Transportation, Healthcare, Smart Environment, and Personal & Social [10] or Personal & Home, Enterprise, Utilities and Mobile [11]. The network characteristics vary widely between the different domains, and even within each domain, the variety of characteristics is high. For example, both smart homes and industrial plants can be considered to be a part of the Smart Environment domain[10]. Still, the network established in a smart home may differ vastly from the network of an industrial farming facility. The floor space of a single-family house is much smaller than that of an industrial plant, and IoT home applications may be focused on human interaction, while industrial IoT is focused on sensing reporting, and adjusting autonomously[3]. Among other things, this implies different network sizes and traffic patterns.

Therefore, while grouping by application domain is useful to map the impact of the IoT and its possibilities for interoperation and interdisciplinary collaboration, this approach to categorizing the IoT is not feasible when it comes to network modeling. Instead, IoT networks can be categorized along a number of characteristics, such as the following:

Traffic Patterns: Some networks experience bursty traffic caused by outside events. Others have a regular, scheduled stream of sensor data. Yet others employ a request/response-cycle based on outside events or internal calculations. Packets may travel towards a central *sink node* in a multipoint-to-point fashion, or flooding the network from a central node as point-to-multipoint traffic, or simple point-to-point. The data rate is typically small.

Mobility: Some or all nodes of a network may experience movement. This can be either constant or shortlived and caused by displacement. Depending on a node's "host", different movement patterns may emerge, such as high speed movement along a fixed grid of streets or slower movements, on less fixed paths, through a more dense space.

Energy-efficiency requirements: Some IoT devices may be built-in to a host with a constant energy supply and therefore not constrained by battery. Other nodes can and will be charged regularly, while yet others must run without maintenance for years.

Size: Maximum Network sizes can range from up to 250 to more than 10.000 nodes, depending on the use case [2] [3] [4]. Some networks, like building automation deployments, are broken down into several subnets containing up to 250 nodes each.

Physical environment: Environmental factors such as walls, nearby objects or weather conditions may impact the node's transmission range [7] and communication behavior.

Depending on these characteristics, networks differ in their requirements for a routing protocol. For example, a building automation installation in a factory might feature 1000 nodes arranged in a star topology with scheduled multipoint to point traffic, no mobility, and high energy efficiency requirements, as nodes are expected to operate on one battery for 5 years[3]. A routing protocol suitable for this environment has lower requirements for latency and code size, but its high energy-efficiency requirements call for high route stability and reliability. On the other hand, a solution monitoring the insides of a food truck features a mesh topology made necessary by the high density of the truck's contents which result in low radio ranges and bursty traffic and node mobility whenever the goods are unloaded or rearranged [7]. However, these goods are stored and monitored in boxes, which could be recharged upon arrival, lowering energy efficiency requirements. Thus, an optimal routing protocol for this environment differs vastly from the protocol suitable for the building automation installation described above: While energy-efficiency is less important here, code and storage size is a relevant factor, since the nodes installed should be as cheap and lightweight as possible. Because boxes can be rearranged during unloading/reloading, timely failure recovery is necessary.

In general, Routing Protocol performance metrics for the IoT can be summarized as:

Latency: The latency with which routes are found or packets are sent can be crucial to some applications. Networks with high mobility may require quick route establishment and usage before the connection is disrupted.

Failure recovery: Especially in highly mobile networks, route disruptions should be recognized and– if possible– fixed in a timely manner.

Route stability: Networks with frequently changing routes can be expensive both in terms of latency as well as battery usage: unless constant routing information is maintained[2], route rediscoveries require increased activity of the transceiver, which is the most battery-hungry component of IoT nodes.

Route Convergence: The longer it takes for a route from one node to the other to be stable, the more bandwidth and energy goes to waste. Additionally, the ability to adapt to mobility is limited.

Code & storage size: With the exception of border routers and sink nodes, IoT devices typically have constrained memory storage resources. Devices which are used in bulk as "throwaway hardware" for monitoring even more so than devices embedded into objects. Protocol complexity and thus code size can be relevant criteria. Another factor is memory usage on operation: extensive routing tables, such as those maintained by proactive protocols, can become a problem

[2] which is the case with most proactive protocols like OLSR[12]

especially in large-scale networks, since they increase linearly with the network size.

Energy consumption: Sending and receiving data is very battery-consuming, so it is advised to keep control traffic as low as possible. Additionally, low handling complexity will help keep retain a high energy efficiency.

Reliability: Routes which experience a high amount of packet loss are prone to triggering packet retransmissions (effectively draining batteries) or losing valuable data. Therefore, it should be ensured that the most reliable route is chosen. A significant part in this is played by *route metrics*, the traits by which a protocol decides which link or route to use. Popular metrics include Hop Count and Expected Transmission Count (ETX).

Based on these characteristics and metrics, an experiment design will be presented in section 5.

4. EXPERIMENT GOALS

It is assumed that all involved routing protocols are fully functional, but excel in different environments. The main goal is not to test them for functionality, but to examine which protocol performs best under which circumstance, and which factors impact routing protocol operations negatively. These factors may be unforeseen quirks which did not occur during previous simulations, or specific network configurations, or something completely different. Protocol performance is assessed using the metrics listed in section 3.

5. EXPERIMENT DESIGN

After all prerequisites have been discussed, a specific experiment can be designed.

5.1 Network model

It can be seen in section 3 that the IoT is a very heterogeneous field in terms of network characteristics, and that a one-scenario-fits-all approach to studying IoT routing is unlikely to be feasible. Therefore, a specific scenario will be studied, modeled and modified in detail over the course of this paper, with the hope that some of the building blocks may be reused as research expands. To achieve this, the characteristics listed as *Default* in table 1 have been chosen as the base scenario to be modeled, as they can be found in a wide range of applications, and are among the most challenging for routing protocols. In order to study which protocol excels in which scenario, variations have to be created. Therefore, some variables of the **Default** scenario are exchanged with an **Alternative** configuration per experiment round.

5.2 Choosing the testbed

In order to run the experiments in a realistic, but still controlled environment, a testbed is needed. Ideally, a testbed suitable for the IoT should be able to provide their users with at least several hundreds, but ideally several thousands of nodes, a diverse range of hardware, and a number of mobile nodes. [5] compares several testbeds with regard to suitability for the IoT, and concludes that the FIT-IoTLab[3] is one of the most suitable facilities. Located all over France, the FIT-IoTLab offers 2,728 nodes in total, featuring three different hardware platforms of different capabilities:

The WSN430 Node featuring a MSP430 MCU with 48kB Flash, 10kB RAM, an IEEE 802.15.4. radio interface, as well as sensors for ambient sensor light and temperature.

The M3 Node featuring an ARM Cortex M3 MCU with 64kB RAM, an IEEE 802.15.4. radio interface, as well as sensors for ambient sensor light, atmospheric pressure and temperature, a gyrometer, and an accelerometer.

The A8 Node featuring an ARM Cortex-A8 microprocessor with 256 MV RAM, an ethernet interface, a gyrometer, and an accelerometer.

Additionally, the IoT-Lab offers node mobility through a fleet of toy trains. This allows for the use of controllable mobility patterns. The two more constrained platforms of the IoT-Lab, the Wsn430 and M3 nodes, offer support for RIOT[13][4]. This combination is unique among all available testbeds, and provides every feature needed to conduct the described experiments. Therefore, it is advised to run the experiments described in this paper on the FIT-IoTLab testbed.

5.3 Routing protocols to test

RIOT currently features implementations of two routing protocols: RPL [14] and AODVv2 [15]. The former is a proactive, point-to-multipoint-protocol designed for LLNs, while the latter is a reactive point-to-point protocol designed for MANETs. Additionally, implementations of the proactive MANET protocol OLSR[12] and the Ant Routing Algorithm (ARA)[16] are in progress.[5] All protocols vary vastly in their characteristics and application scenarios, so it would be advisable to involve as many as possible in the experimentation. In addition to RPL and AODVv2, any other protocols available by the time of the experiment should be used.

6. SETUP AND EXECUTION

Now that all metrics and environmental variables have been determined, the appropriate experiment setup and execution can be discussed. In preparation of the experiment, the following is created:

1. To create multipoint-to-point traffic: a list containing the IDs of all *except for one* participating nodes.
2. To create point-to-point traffic: a randomized list containing tuples with randomized pairings of the IDs of all participating nodes. It should contain some duplicates and be of length `max_transmissions`.
3. A sample packet with a payload of 20 bytes, resulting in a 61-byte packet including IEEE 802.15.4. and IPv6 headers with applied 6LoWPAN header compression.
4. A randomized, duplicate-free list of length `num_failing_nodes`, containing IDs of participating nodes.

These lists must never be changed throughout the whole experiment, and should be stored along with the experiment data for future reference.

[3] `https://www.iot-lab.info`

[4] `https://www.iot-lab.info/operating-systems/`, accessed 19.05.2015

[5] `https://github.com/RIOT-OS/RIOT/pull/2294`
`https://github.com/mfrey/RIOT/tree/ara`

Characteristic	Default	Alternative
Traffic Pattern	Multipoint-to-point, with most traffic traversing several hops. Scheduled data transmissions.	point-to-point across the network. Scheduled data transmissions.
Mobility	None, but occasionally failing nodes.	–
Energy efficiency reqs.	None	–
Network size	100	500
Physical environment	IoT-Lab testbed	–

Table 1: Characteristics of the modeled network(s)

One experiment run consists of multiple sub-experiments called *scenarios*, all of which are repeated for every routing protocol involved. As discussed in section 5.1, routing protocols should be tested in different environments to explore under which conditions they excel. Thus, the default network model presented in table 1 and its variations each are the base for one scenario. This creates a total of four scenarios which make up one experiment:

Scenario 1: default characteristics only
Scenario 2: alternative size
Scenario 3: alternative traffic pattern
Scenario 4: alternative size and traffic pattern

Each experiment run is conducted as follows: To emulate failing batteries, nodes are shut down every `max_transmissions/num_failing_nodes`'th iteration in each scenario.

For scenarios 1 and 2, List number 1 is used to model multipoint-to-point traffic. The node not contained in the list is appointed as the *sink node* towards which all traffic is directed. Then, the list is traversed sequentially to initiate sending. There is a waiting interval of `fixed_time` between each two transmissions. [6]

```
for t in range (0, max_transmissions):
  s = max_transmission/num_failing_nodes
  if (t % s == 0 ):
    failing_nodes.pop().shutdown()
  for node in node_ids:
    node.send_packet(sink_node)
sleep(fixed_time)
```

Note that all nodes send at roughly the same scheduled time and then sleep collectively.

For scenarios 3 and 4, List number 2 is used to model point-to-point traffic. Each tuple `(node_1, node_2)` represents a transmission from `node_1` to `node_2`. This list too is traversed sequentially to initiate sending:

```
for t in range (0, max_transmissions):
  s = max_transmission/num_failing_nodes
  if (t % s == 0 ):
    failing_nodes.pop().shutdown()
  for (node_1, node_2) in node_ids:
      node_1.send_packet(node_2)
      sleep(fixed_time)
```

This ensures the exact same transmission sequence for each experiment run, eliminating possible side effects. For each scenario combination, each experiment is run a fixed number of `max_experiment_runs` times, so as to not eschew data by isolated incidents.

The variables used above should be substituted for actual values as follows:

variable	scenarios 1 & 2	scenarios 3 & 4
`fixed_time`	5 seconds	1 second
`max_transmissions`	$size \cdot 2$	$size \cdot 2$
`max_experiment_runs`	30	30
`num_failing_nodes`	10% of all nodes	10% of all nodes
`packet size`	61 bytes	61 bytes

Table 2: Experimentation values

All numbers used should be tweaked in case they are found to be unrealistic or statistically problematic.

During each experiment run, the following data is collected per node in a machine-readable format, along with a timestamp:

- Each routing table update.
- Each sent data packet.
- Each received data packet.
- Each sent control packet.
- Each received control packet.
- Overall energy consumption.

Additionally, the overall size of the RIOT image is recorded as the code and storage size. Since time synchronization in a network is a complicated problem, close attention should be paid to the accuracy of the collected timestamps. If the margin of error between the nodes of the network is not negligible, an alternate solution should be investigated.

6.0.1 Experiment implementation

In order to provide a reusable and tweakable setup, modularity is to be kept in mind when implementing this experiment. Components should be parametrized wherever possible to allow further experiment variation. Additionally, the resulting experiment setup should be easy to use to enable the reproduction of experiments.

6.1 Experiment evaluation

The success or failure of each routing protocol is determined by the metrics listed in section 3. A protocol's performance regarding a certain metric is evaluated as follows:

Latency: The median difference between packet dispatching and arrival time is used to calculate the latency

[6] All pseudocode is based on python syntax.

with which packets are sent. For proactive protocols, the median time in which routes appear in the Forwarding Information Base (FIB) is taken for route finding latency. For reactive protocols, each yet unknown tuple is examined: the median time it takes between sending attempt and appearance of the route in the FIB determines route creation latency.

Failure recovery: The median time from node shutdown to changes in routing table and FIB is calculated.

Code & storage size: Memory usage is monitored during the experiment. The median as well as the maximum memory usage are used to determine memory efficiency, as well as the code size at compile time.

Energy-efficiency: Energy usage is monitored during the experiment as well and can be compared between protocols per experiment batch.

Reliability: Packets are recorded when they are sent at one end and received at the other. This way, the median number of lost packets can be determined.

Apart from the overall median, the median value of the above should also be calculated over all experiment runs at each point in time and be plotted into graphs to detect any difference in development. For example two protocols with about the same overall median latency might show punctual differences in latency after a disruption of the network.

7. CONCLUSION AND OUTLOOK

Over the course of this paper, the necessity of IoT routing protocol experimentation aided by testbeds has been discussed. Challenges in the creation of a setup have been discussed, along with possible solutions and a concrete experimentation setup. It has been stressed that this setup is to be extended to fully honor the diversity of IoT scenarios, and that it can be merely a starting point. These extensions may not only be limited to further switching of parameters and increasing the network size. Once initial experience is gathered, sparse mobility should be added to the experiments. Instead of waiting for a fixed time interval between transmissions, the waiting time could be randomized, or physically close nodes could send simultaneously, simulating a local triggering event. Detailed mobility schemes could be developed, or more complex and/or hybrid traffic patterns. A wider range of protocols should be tested: protocols with similar characteristics could be compared against each other, or the same protocol could be tested with different route metrics.

Based on the findings of all of these experiments, a map of protocol characteristics suitable for different IoT scenarios can be created. Future work could also include the development of optimizations or extensions targeting specific scenarios for any of the routing protocols involved based on the findings gained through testbed evaluation.

Before all of this can be done, however, the provided experimentation scenarios will have to be implemented, put to the test, and tweaked.

8. REFERENCES

[1] M. S. Corson and J. Macker, “Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations,” RFC 2501, IETF, January 1999.

[2] A. Brandt, J. Buron, and G. Porcu, “Home Automation Routing Requirements in Low-Power and Lossy Networks,” RFC 5826, IETF, April 2010.

[3] J. Martocci, P. D. Mil, N. Riou, and W. Vermeylen, “Building Automation Routing Requirements in Low-Power and Lossy Networks,” RFC 5867, IETF, June 2010.

[4] M. Dohler, T. Watteyne, T. Winter, and D. Barthel, “Routing Requirements for Urban Low-Power and Lossy Networks,” RFC 5548, IETF, May 2009.

[5] A.-S. Tonneau, N. Mitton, and J. Vandaele, “A survey on (mobile) wireless sensor network experimentation testbeds,” in *Distributed Computing in Sensor Systems (DCOSS), 2014 IEEE International Conference on*, pp. 263–268, May 2014.

[6] K. Heurtefeux and H. Menouar, “Experimental evaluation of a routing protocol for wireless sensor networks: Rpl under study,” in *Wireless and Mobile Networking Conference (WMNC), 2013 6th Joint IFIP*, pp. 1–4, April 2013.

[7] R. Jedermann, T. Pötsch, and C. Lloyd, “Communication techniques and challenges for wireless food quality monitoring,” *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 372, no. 2017, 2014.

[8] D. Kotz, C. Newport, and C. Elliott, “The mistaken axioms of wireless-network research,” tech. rep., Dartmouth Computer Science, July 2003.

[9] G. Coulson, B. Porter, I. Chatzigiannakis, C. Koninis, S. Fischer, D. Pfisterer, D. Bimschas, T. Braun, P. Hurni, M. Anwander, G. Wagenknecht, S. P. Fekete, A. Kröller, and T. Baumgartner, “Flexible experimentation in wireless sensor networks,” *Commun. ACM*, vol. 55, pp. 82–90, Jan. 2012.

[10] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” *Comput. Netw.*, vol. 54, pp. 2787–2805, Oct. 2010.

[11] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of things (iot): A vision, architectural elements, and future directions,” *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645 – 1660, 2013.

[12] T. Clausen and P. Jacquet, “Optimized Link State Routing Protocol (OLSR),” RFC 3626, IETF, October 2003.

[13] E. Baccelli, O. Hahm, M. Günes, M. Wählisch, and T. C. Schmidt, “RIOT OS: Towards an OS for the Internet of Things,” in *Proc. of the 32nd IEEE INFOCOM. Poster*, (Piscataway, NJ, USA), IEEE Press, 2013.

[14] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, “RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks,” RFC 6550, IETF, March 2012.

[15] C. Perkins, S. Ratliff, J. Dowdell, L. Steenbrink, and V. Mercieca, “Dynamic MANET On-demand (AODVv2) Routing,” Internet-Draft – work in progress 09, IETF, May 2015.